**目录**

[1. 前言 1](#_Toc455038194)

[2. 升级前注意 1](#_Toc455038195)

[3. 升级单个mongod实例 2](#_Toc455038196)

[4. 升级副本集 2](#_Toc455038197)

[5. 升级分片集群 4](#_Toc455038198)

[5.1升级前注意 4](#_Toc455038199)

[5.2关闭均衡器 5](#_Toc455038200)

[5.3备份并更新元数据 6](#_Toc455038201)

[5.3.1备份元数据 6](#_Toc455038202)

[5.3.2关闭40002端口的mongos用其更新元数据 6](#_Toc455038203)

[5.4更新其他组件 7](#_Toc455038204)

[5.4.1更新mongos 7](#_Toc455038205)

[5.4.2更新mongod-config组件 7](#_Toc455038206)

[5.4.3更新分片副本集 7](#_Toc455038207)

[6. 再说两句 10](#_Toc455038208)

[7. 参考资料 10](#_Toc455038209)

**jar cvfm MyWordCount.jar manifest.data \*.class**

**关键数据升级务必进行升级安全评估，避免升级过程出现问题对业务造成致命影响！！！**

# 前言

和前面升级类似，能直接升级到2.6的版本是2.4，其他低版本必须依次进行，本篇介绍**如何从2.4升级到2.6**。

# 升级前注意

1. 用2.6版本的mongo连接到2.4版本的mongod或者mongos上面，运行如下命令：

**use admin;**

**db.upgradeCheckAllDBs();**

如果你不在admin下运行或者使用2.4版本执行，会出现错误。

成功运行后，该命令会检查所有的数据是否符合更新条件，如果都可以如下输出：



如果上面的检查出现了问题，必须解决所有问题再进行升级。另外，还有一些兼容性问题需要你手动检查，这些兼容性点参考2.6兼容性https://docs.mongodb.com/manual/release-notes/2.6-compatibility/。

1. 如果2.4的部署使用了身份认证，那么必须确认在admin database中至少存在一个**拥有所有权限（role userAdminAnyDatabase，可以操作任意数据库）**的用户；
2. 2.6使用了新的用户认证模型，在更新完所有的MongoDB组件之后，需要更新用户认证模型（参考：更新2.6的用户认证模型），否则你将无法修改user data。

**基于这个考虑，我们可以考虑在业务设计前期使用认证访问和非认证访问两种方式进行，如果升级认证模型太过繁琐，我们可以将其删除，直接采用非认证模式跳过这个过程。**

1. 注意一旦升级到2.6，只能降级到2.4.10+的版本；

# 升级单个mongod实例

只能从2.4升级到2.6。

（1）在<https://www.mongodb.com/download-center#community>下载2.4版本的MongoDB；

（2）关闭旧的mongod，然后用2.6版本的MongoDB直接重启；

**J:\mongodb-64-2.4.8\bin\mongo ip:port**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --port=port --dbpath="…"**

# 升级副本集

同样，只能从2.4更新到2.6版本。这个过程和2.2升级到2.4类似，可以进行热升级。

为了方便描述，我们假定升级如下模型的副本集：



1. 首先更新各个secondary结点（这是必须的），我们先关闭Secondary-2结点的mongod，然后用mongod2.6替换重启。等待Secondary-2恢复到Secondary状态，再继续进行；

**J:\mongodb-64-2.4.8\bin\mongo ip:port**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到Primay结点执行rs.status()查看结点状态确认已经停掉**

**J:\mongodb-64-2.6.12\bin\mongod --port=port --dbpath="…" --replSet <replSetName>**

1. 等到Secondary-2恢复到Secondary状态之后，我们用同样的方法升级一下Secondary-1结点；
2. 此时，我们切换Primary到另外两个结点中的一个。登录到Primary上，执行切换动作，然后用2.6重启该节点；

**J:\mongodb-64-2.4.8\bin\mongo ip:port**

**use admin;**

**rs.stepDown(); # 切换Primay结点，这个动作会强迫执行一个故障转移failover**

**db.shutdownServer(); # 如果不放心，执行rs.status()查看结点状态确认已经关闭**

**J:\mongodb-64-2.6.12\bin\mongod --port=port --dbpath="…" --replSet <replSetName>**

由此可见，副本集的更新到2.6时是允许热更新的，而且注意到允许存在不同版本的mongod实例运行。

副本集的故障转移（failover，失效备援、故障转移）并不是立即完成的，在故障转移完成之前是不能进行写操作的，通常需要等待30s或者更多。

# 升级分片集群

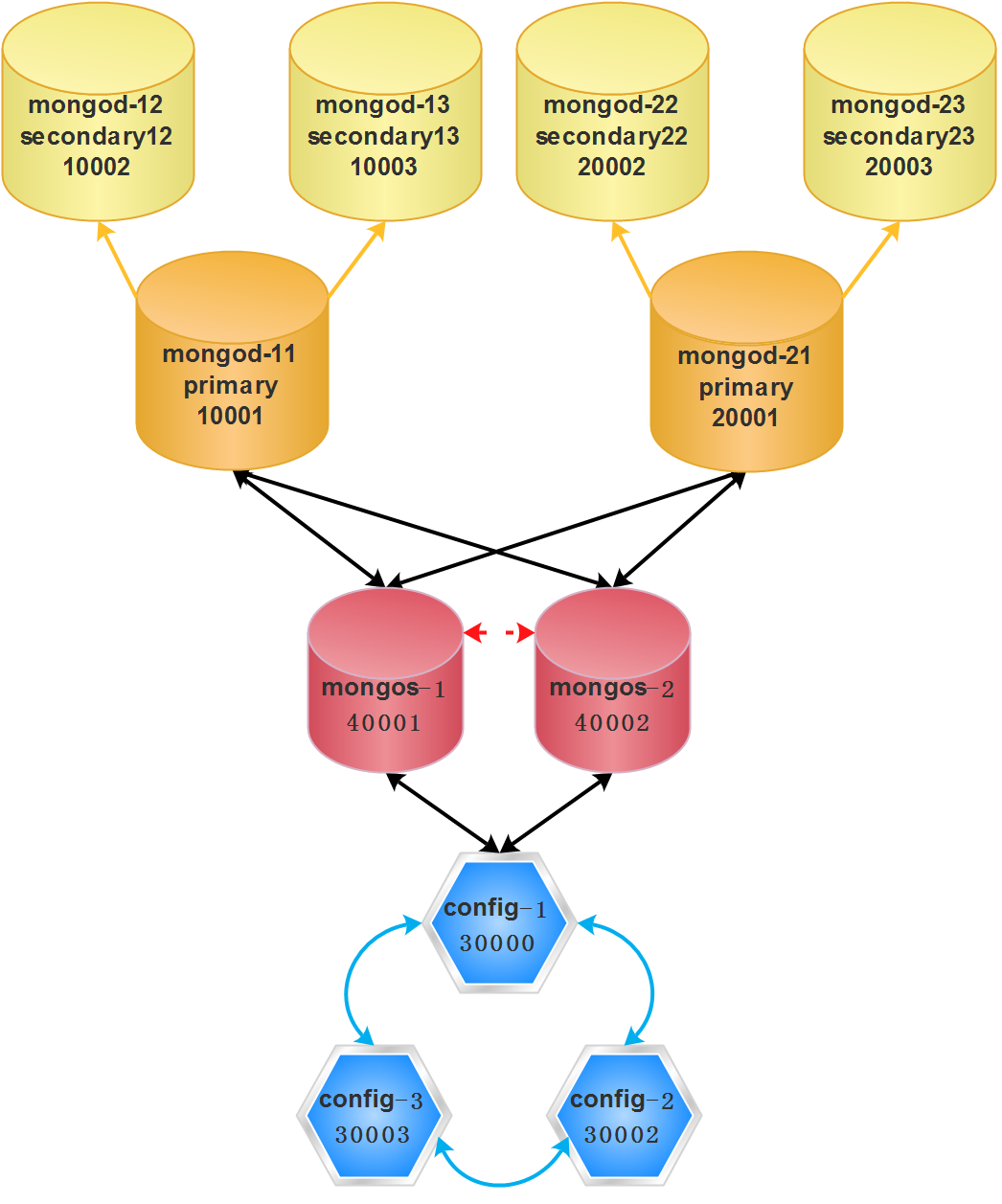
同样只能从2.4集群升级到2.6，不能是更低版本。

## 5.1升级前注意

升级集群可以不下线，但是必须确保在升级过程中不能有修改元数据的操作，比如：

* [sh.enableSharding()](https://docs.mongodb.com/manual/reference/method/sh.enableSharding/#sh.enableSharding)；
* [sh.shardCollection()](https://docs.mongodb.com/manual/reference/method/sh.shardCollection/#sh.shardCollection)；
* [sh.addShard()](https://docs.mongodb.com/manual/reference/method/sh.addShard/#sh.addShard)；
* [db.createCollection()](https://docs.mongodb.com/manual/reference/method/db.createCollection/#db.createCollection)；
* [db.collection.drop()](https://docs.mongodb.com/manual/reference/method/db.collection.drop/#db.collection.drop)；
* [db.dropDatabase()](https://docs.mongodb.com/manual/reference/method/db.dropDatabase/#db.dropDatabase)；
* any operation that creates a database；
* <https://docs.mongodb.com/manual/reference/sharding/>这个地方的所有命令，注意并不是所有命令都会导致修改元数据，但是为了确保最好在升级元数据的过程不要使用这个链接出的所有命令；

更新过程同样分为：元数据更新、其他组件更新。假定我们要更新的集群模型如下：



详细更新过程如下：

## 5.2关闭均衡器

登录到40001端口的mongos上关闭均衡器，如下操作：

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:40001**

**MongoDB shell version: 2.4.8**

**connecting to: 127.0.0.1:40001/test**

**mongos> sh.stopBalancer();**

**Waiting for active hosts...**

**Waiting for the balancer lock...**

**Waiting again for active hosts after balancer is off...**

**mongos> sh.getBalancerState();**

**false**

**mongos>**

## 5.3备份并更新元数据

### 5.3.1备份元数据

首先我们备份配置数据库的元信息：

**J:\mongodb-64-2.4.8\bin\mongodump -host 127.0.0.1:30001 -d config -o "G:\mongo升级过程临时目录\1 备份配置数据库mongo-config"**

### 5.3.2关闭40002端口的mongos用其更新元数据

注意，升级元数据的时候，mongod-config的服务一定要处于开启状态，否则升级谁呢，对吧！

我们的集群中有2个mongos，我们逐个升级，这样可以避免业务受牵连（前提是业务采取了轮询mongos的方式访问）。在此我们先关闭40002端口的mongos服务，利用40002端口的mongos服务来升级元数据，如下进行。

**# 关闭旧的40002端口mongos**

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:40002**

**use admin;**

**db.shutdownServer();**

**# 在40002端口启动一个新2.6版本的mongos加上--upgrade用来更新元数据**

**J:\mongodb-64-2.6.12\bin\mongos.exe --port=40002 --configdb="127.0.0.1:30001,127.0.0.1:30002,127.0.0.1:30003" --upgrade**

执行更新命令之后如果输出如下信息：

**2016-06-29T16:06:42.639+0800 [mongosMain] upgrade of config server to v5 successful**

**2016-06-29T16:06:42.776+0800 [mongosMain] distributed lock 'configUpgrade/LIULEI26:40002:1467187600:41' u**

**nlocked.**

**2016-06-29T16:06:42.777+0800 [mongosMain] Config database is at version v5**

则表示更新成功，在此处略和2.2更新到2.4不同的是：2.2更新到2.4时更新完成mongos不会关闭，而此处mongos会关闭，我们进行重启即可。

**J:\mongodb-64-2.6.12\bin\mongos.exe --port=40002 --configdb="127.0.0.1:30001,127.0.0.1:30002,127.0.0.1:30003"**

## 5.4更新其他组件

### 5.4.1更新mongos

刚才我们更新了40002端口的mongos，现在我们更新40001端口的mongos，此时就不要--upgrade选项了，如下操作即可：

**# 关闭旧的40001端口mongos**

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:40001**

**use admin;**

**db.shutdownServer();**

**# 在40001端口启动一个新2.6版本的mongos用来更新元数据**

**J:\mongodb-64-2.6.12\bin\mongos.exe --port=40001 --configdb="127.0.0.1:30001,127.0.0.1:30002,127.0.0.1:30003"**

### 5.4.2更新mongod-config组件

我们依次更新3个mongod-config服务。按照官方建议按照mongos --configdb后面参数顺序从后往前更新，我们的更新顺序是30003、30002、30001。

* 关闭并重新启动30003端口的mongod-config服务

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:30003**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --dbpath="G:\mongodb-config3-30003\data" --port=30003 --configsvr**

* 关闭并重新启动30002端口的mongod-config服务

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:30002**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --dbpath="G:\mongodb-config2-30002\data" --port=30002 --configsvr**

* 关闭并重新启动30001端口的mongod-config服务

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:30001**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --dbpath="G:\mongodb-config1-30001\data" --port=30001 --configsvr**

至此，所有的mongod-config配置服务器已经更新完毕。

### 5.4.3更新分片副本集

至此我们的更新操作已经完成了一大半了，下面我们来更新分片集群。我们依次更新lt01、lt02分片。

关键点：先切换Secondary结点，然后更替Primary结点，最后再切换之前的Primary结点。

* 更新lt01分片

更新分片时稍微麻烦一点，有一个Primary结点更替的命令需要执行。因为我们的分片Primary承担着数据写入的任务，所以Primay必须时刻存在才能不影响业务。我们按照：lt01-10003、lt01-10001、lt01-10002的顺序依次更新，因为当前10002为Primay结点，因此不能第一个更新它。

1. 停止lt01-10003端口并以mongod2.6重启

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:10003**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到10001结点执行sh.status()查看结点状态确认已经停掉**

**J:\mongodb-64-2.6.12\bin\mongod --port=10003 --dbpath="G:\mongodb-shard01repl-mongod3-10003\data" --replSet=lt01 --shardsvr**

1. 停止lt01-10001端口并以mongod2.6重启

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:10001**

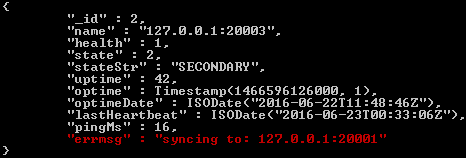
**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --port=10001 --dbpath="G:\mongodb-shard01repl-mongod1-10001\data" --replSet=lt01 --shardsvr**

1. 切换Primary结点

一定要注意！！！在切换之前查看另外两个已经启动的副本结点状态，rs.status()查看状态之后，不要出现errmsg的信息为“syncing to: \*\*\*\*\*\*“如下，



如果errmsg有如上信息说明此时正在从primary同步数据，此时最好等待完成，在进行结点的切换（官方没有，测试时个人发现，觉得应该这么做）。

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:10002**

**rs.stepDown()**

**rs.status();**

另外，还要注意，切换完结点之后要确认查看一下10002当前的状态，确认已经不是PRIMAYR结点了并且另外两个结点有一个已经是PRIMARY结点了，并且rs.status()之后没有errmsg或者syncing to\*\*\*的相关信息。

1. 停止lt01-10002端口并以mongod2.6重启

Primary结点已经切换，此时我们可以更新lt01-10002端口的结点了：

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:10002**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --port=10002 --dbpath="G:\mongodb-shard01repl-mongod2-10002\data" --replSet=lt01 --shardsvr**

至此lt01分片更新完成。

* 更新lt02分片

更新分片lt02同lt01，lt的Primay结点是20001端口的mongod，因此我们按照20003、20002、20001的顺序更新，操作如下：

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:20003**

**use admin;**

**db.shutdownServer(); # 如果不放心，可以登录到20001结点执行sh.status()查看结点状态确认已经停掉**

**J:\mongodb-64-2.6.12\bin\mongod --port=20003 --dbpath="G:\mongodb-shard02repl-mongod3-20003\data" --replSet=lt02 --shardsvr**

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:20002**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --port=20002 --dbpath="G:\mongodb-shard02repl-mongod2-20002\data" --replSet=lt02 --shardsvr**

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:20001**

**rs.stepDown()**

**rs.status(); # 查看一下20001当前的状态，确认已经不是PRIMAYR结点了**

**J:\mongodb-64-2.4.8\bin\mongo 127.0.0.1:20001**

**use admin;**

**db.shutdownServer();**

**J:\mongodb-64-2.6.12\bin\mongod --port=20001 --dbpath="G:\mongodb-shard02repl-mongod1-20001\data" --replSet=lt02 --shardsvr**

至此，整个集群更新完成。相比从2.2.1+更新到2.4，此次更新略显简单。

# 再说两句

本篇暂且略过用户认证模式的更新，关于用户认证模式的更新参考此处：https://docs.mongodb.com/manual/release-notes/2.6-upgrade-authorization/。

# 参考资料

1. https://docs.mongodb.com/manual/release-notes/2.6-upgrade/
2. https://docs.mongodb.com/manual/release-notes/2.6-upgrade-authorization/